# Capítulo 8 - Análisis y selección de tecnologías para desarrollo del SAR

En capítulos anteriores analizamos diversas tecnologías hardware y software relacionadas con la robótica. Dentro de las tecnologías hardware investigamos las plataformas Arduino (Capítulo 3 – Arduino) y Raspberry Pi (Capítulo 4 – Raspberry Pi). De las diversas herramientas en tecnologías software para aplicaciones móviles, investigamos aquellas que permiten desarrollar una aplicación que interactúe con el hardware mencionado.

En este capítulo examinamos dichas tecnologías para concluir cuales son las que integran el SAR. Para ello, realizamos diversas comparativas sobre características, ventajas y desventajas de cada una de las plataformas. Además, se presentan las problemáticas surgidas al relacionar los componentes.

## **8.1 Primer análisis**

El análisis realizado de las distintas tecnologías de hardware y software, basándose en factores como rendimientos, tiempos de respuesta, consumo energético, portabilidad y compatibilidad, nos permitió seleccionar los elementos que componen el SAR.

El hardware estudiado y utilizado a lo largo del desarrollo, fue el de la familia Arduino y Raspberry Pi.

Dentro de la plataforma Arduino se probaron las versiones Arduino Uno, Arduino Mega y Arduino Nano, siendo estos dos últimos los utilizados en el SAR. Además, se experimentaron con variados módulos, sensores y actuadores compatibles con esta familia como la cámara OV7670, ESP8266, Bluetooth, DHT11, entre otros. Los ensayos realizados con estos componentes se encuentran en el anexo de casos pruebas de módulos, sensores y actuadores.

En el caso de Raspberry Pi utilizamos la versión Pi 3 modelo B, en conjunto con la cámara compatible para esta SBC.

El software investigado para el desarrollo de aplicaciones móviles fue mencionado en los capítulos 5 y 6 (Capítulo 5 - Aplicaciones Móviles y Capítulo 6 – Stack MEAN). La idea era encontrar compatibilidad entre los elementos del hardware, utilizando el software como interfaz entre ellos.

## **8.2 Selección tecnologías hardware**

### 8.2.1 Razones para la elección de Arduino

Como se abordó en el capítulo 3 (Capítulo 3 – Arduino), siendo una arquitectura hardware pensada para hobbistas, diseñadores y personas no relacionadas con la electrónica ni la programación a bajo nivel, Arduino permite una curva de aprendizaje relativamente baja y la disponibilidad componentes conectables lo hace muy atractivo, para afrontar proyectos con diversos niveles de complejidad.

Gracias a que existe buen soporte de placas Arduino para el uso en control mediante en sensores y actuadores basado en microcontrolador, es atractivo en el contexto de una transición desde electrónica discreta hacia la programable.

Dentro de la plataforma Arduino, se seleccionó la placa Arduino UNO, sobre la cual se elaboraron distintos prototipos simples con *protoboard*, desde la manipulación de actuadores con motores hasta la toma de datos de distintos sensores como temperatura, humedad, distancia, entre otros. Se encontró en la placa Arduino UNO una baja disponibilidad de pines E/S para la cantidad de sensores/actuadores y módulos que se deseaban conectar. Por tal motivo, se decidió ampliar la cantidad de pines optando por la placa Arduino Mega. Esta última, otorga mayor cantidad de pines, sin expandir la cantidad de memoria ni procesamiento. Este detalle devino en la dificultad a la hora de la programación, por contar con pocas interrupciones hardware, forzando a la utilización de consulta periódica o *pooling* en el bucle principal (loop).  Estos problemas surgieron a la hora de conectar el módulo de la cámara OV7670 y el módulo WiFi ESP8266, los cuales requerían una alta cantidad de pines y nivel de cómputo.

A razón de estos problemas, se optó por el transpaso de una plataforma que trabaja con un microcontrolador a una basada en un computador, siendo elegida la Raspberry Pi.

### 8.2.2 Razones para la elección de Raspbery Pi

En el capítulo 4 (Capítulo 4 – Raspberry Pi) se mencionó y analizó el computador de placa reducida (SBC) Raspberry Pi, plataforma diseñada primordialmente con fines didácticos por lo que su costo es relativamente bajo. Al contar con todas las capacidades básicas de una computadora portátil de hoy en día con su respectivo microprocesador (de potentencia suficiente para las necesidades del SAR), memorias y puertos físicos (como el USB, HDMI, microSD, entre otros); y la posibilidad de instalar un sistema operativo totalmente funcional y con interfaz gráfica (en este caso Raspbian), es que se seleccionó como centro de administración y control del SAR.

Además, cuenta con pines GPIO para las conexión y manipulación de distintos módulos (como actuadores y sensores), aunque como se analiza en el apartado siguiente, se delegó en placas Arduino Mega y Arduino Nano las funcionalidades de control y sensado, exceptuando la conexión y procesamiento de imágenes, delegadas a la cámara de Raspberry Pi v2 y las comunicaciones inalámbricas proporcionadas por los módulos wifi y bluetooth integrados a este computador.

### 8.2.3 Comparativa entre Arduino Mega, Arduino Nano y Raspberry Pi 3 Model b

|  |  |  |  |
| --- | --- | --- | --- |
| Factor | Arduino Mega | Arduino Nano | Raspberry Pi3 Model B |
| Microcontrolador/  Microprocesador | ATmega 1280 - 16Mhz 8bits | ATmega328 – 16Mhz 8bits | Quad Core 1.2GHz Broadcom BCM2837 64bit |
| Tensión | 5v | 5v | 5v |
| Memoria | 128 KB (Bootloader 4KB) | 32 KB (Bootloader 2KB) | 1 GB |
| Digital I/O | 54, 15 PWM | 22, 6 PWM | 40 GPIO |
| Analog I/O | 16 | 8 |
| Interfaces | USB x 1(energía) | USB x 1 (energia) | USB x 4, HDMI, CSI, DSI, MicroSD, WLAN y BLE, microUSB (Energía) |

Dada la comparativa entre las tecnologías, se decide utilizar las placas Arduinos (versiones Nano y Mega) para el control de módulos de hardware como sensores y actuadores y la SMB Raspberry Pi dedicada la captura de imágenes y ejecución de servidor web. La conexión entre Arduino Raspberry a través de sus interfaces USB.

¿Cuáles son los beneficios de esta arquitecura? Si bien podría pensarse que se sería suficiente Raspberry para la elaboración del SAR, deben considerarselos siguientes beneficios que proporciona Arduino:

* Menor costo del producto para sustitución ante fallos.
* Mayor compatibilidad, con los módulos arduino-compatibles (como los de la familia Adafruit)
* Buen tiempo de respuesta para I/O.
* Alta confiabilidad en la lectura de sensores y en los valores de manipulación de actuadores.

Varios de estos beneficios se deben a que Arduino no posee un sistema operativo, sino un único programa que se ejecuta indefinidamente (LOOP) sin necesidad de correr algún software auxiliar que lo dispare o ejecutando como servicio; logrando concentrar su poder de procesamiento en el único programa definido. Al carecer de un sistemas operativo, no existen retrasos inesperados propios de la artquitectura con protección de memoria y paginación ni tampoco los de la política de pogramación de tareas (*scheduling*).

### 8.2.4 Cámara V2 de Raspberry Pi

Como se mencionó anteriormente, se delegó la tarea de captura de imágenes a Raspberry, a través de la cámara exclusiva de esta plataforma en su versión V2. Esta se conecta al puerto CSI de cualquier modelo de este SBC, lo cual permite obviar la conexión pin a pin (como ocurre con cámaras como la OV7660) y no es necesario controlar la comunicación y captura de imágenes. Como se comentó en el capítulo 4, es una cámara de alta definición de 8 megapíxeles, suficiente para el objetivo que se pretende con el desarrollo del SAR. Esto soluciono las problemáticas que se nos presentaron a la hora de probar la cámara OV7670 con Arduino; como el poder de procesamiento de imágenes y transmisión de las mismas (inalámbricamente) hacia otro dispositivo tal como una PC o un dispositivo móvil (en el caso de esta tesina smartphones).

### 8.2.5 Módulos de Arduino

Dentro de los módulos, sensores y actuadores de Arduino que se probaron y/o se utilizan, se encuentran:

*Utilizados en el sar:*

* El módulo GPS, será utilizado para determinar la ubicación geográfica del SAR (Geolocalización)
* Sensor de temperatura KY-001(-55° a +125°)
* Sensor ultrasonido HC-SR04 para determinar presencia de objetos a determinadas distancia y tratar de evitar el impacto con los mismos
* Motores CC para la movilidad del SAR dentro del ambiente

*Ensayados y no seleccionados:*

* El módulo wifi ESP8266 y el módulo Bluetooth HC-05, no se utilizarán debido a que la Raspberry Pi3 Model B, brinda su funcionalidad.
* El módulo Acelerómetro MMA7361.
* Servomotor sg90.
* Sensor de evasión de obstáculos KY032.
* Sensor de golpe KY-031.
* Sensor de llamas KY-026.

## **8.3 Selección tecnologías software**

### La selección del software, necesario para el desarrollo del SAR, se basa en los siguientes requerimientos:

### Nivel de abstracción alto, logrado mediante librerías basadas en JavaScript, para la comunicación con el hardware (J5).

### Utilizar un Sistema Operativo de base (en nuestro caso Raspbian), en vez de una rutina corriendo en un microcontrolador.

### Contar con recursos necesarios para desplegar un servidor web.

### Disponer de la posibilidad de comunicar las plataformas Arduino al servidor mediante un protocolo bien conocido.

### Utilizar las herramientas de SO del SBC para realizar la comunicación y captura de imágenes por sobre la captura manual de frames.

### Desarrollar de una aplicación móvil para el control inalámbrico del SAR.

### Almacenar datos para la generación de estadísticas

### Permitir el acceso multi-cliente a los datos alojados en él SAR.

### Se optó por la instalación de Raspbian en la Raspberry, porque es el sistema operativo oficialmente soportado por la fundación[[1]](#endnote-1). Como se mencionó en un apartado anterior (8.2.1 ¿Por qué Arduino?) en cuanto a las dificultades que surgieron al tratar de utilizar la cámara OV7670 con el Arduino Mega, es que se decidió adquirir la Raspberry Pi 3 modelo B. Esta plataforma cuenta con un accesorio que funciona como cámara (mencionada en el apartado 4.6 Accesorios para Raspberry Pi) tal se tratase de una webcam.

se encontróestá orientada a el caso del SAR en forma de *streaming*.

Dentro de las dificultades afrontadas con Arduino que devinieron en la delegación de funciones a Raspberry encontramos, que las placas esan orientadas a programas dónde existe un único bucle de ejecución principal, en el caso del control de una camara, nser d, tanto en como en forma inalámbrica (requiriendose shields de comunicación), ndocuadros por segundomínimamente Por otro lado, para poder almacenar gran cantidad de datos es necesario contar con un módulo para memorias SD. Entre otras desventajas de las placas, poseen una cantidad limitada de interrupciones por hardware (2 en Arduino Uno y Nano, 6 en el caso de Arduino Mega), resultando en la detección de nuevo valores en sensores mediante pooling.

Ante estas limitaciones Raspberry gracias a su hardware y ser un computador que permite la instalación de un sistema operativo, facilitó resolver varias de las dificultades antes mencionadas. Se destacan, por ejemplo, las capacidades inalámbricas que permitieron configurar la SBC en modo Access Point (AP, o punto de acceso en español Glosario). Esto quiere decir, crear una red inalámbrica WiFi (con una SSID y contraseña) sin depender de ningún dispositivo de red externo (como por ejemplo un router inalámbrico) y permitiendo la conexión de diversos hosts (Glosario), donde cada uno obtiene su respectiva dirección IP por medio de DHCP.

Al comienzo del desarrollo, y teniendo en cuenta que dónde incialmente se pensó en trabajar únicamente con la familia Arduino, se había pensado en diseñar una aplicación móvil nativa. Dado que la única comunicación que existía entre un posible cliente y el SAR era por datos *raw (Glosario)* enviados por bluetooth o wifi (el compendio de tecnlogías relacionadas con esta App, se abordaron en el Capítulo 5 - Aplicaciones Móviles); pero al mejorarse las prestaciones hardware y tener un sistema operativo, se decidió cambiar la arquitectura del software del SAR.

Esta nueva arquitectura genero un cambio en la aplicación, o sea, se pasó del desarrollo de una app nativa, para Android, a una app Web, premitiendo crear una única aplicación que puede ser consumida por distintos dispositivos que accedan a la red LAN (Glosario) del SAR.

Para producir la app web se necesitó de un grupo de tecnologías que satisfagan los siguientes puntos:

* Contar con la posibilidad de almacenar datos de los sensores y acciones realizadas mediante una base de datos.
* Tener una interfaz de comunicación sencilla con el servidor.
* Tener la capacidad de desplegar a demanda la app desde una red LAN.
* Diseñar una app, utilizando herramientas de frontend, para el renderizado en el cliente, para una mejor experiencia de usuario, basada en requierimientos HTTP para la comunicación con el servidor.

Tanto Cordova como IntelXDK fueron descartadas dado que se prefirió un grupo de herramientas, compatibles entre ella, y estables (stack de desarrollo de software).

Por otro lado, se trató de incursionar en Meteor, realizando aplicaciones sobre arquitecturas Intel x86/x64. Se diseño un prototipo funcional de la aplicación, pero al migrar la misma a la arquitectura ARM (Raspberry Pi) se encontraron inconvenientes dado que este framework no se encontraba soportado oficialmente para esta arquitectura. Existe un *fork*, pero no se tuvo éxito en la integración de las tecnologías.

Finalmente se seleccionó el stack MEAN el cual resultó ser compatible con el desarrollo avanzado hasta el momento, hecho con Meteor (MEAN se encuentra detallado en el Capítulo 6 – Stack MEAN). La migración de la aplicación tanto *frontend* como *backend*, desarrollada con Meteor, fue dispuesta de la siguiente forma: El procesamiento de *templates*, captura y gestión de eventos, realizada en Blaze, se trasladó a Angular 4+. El servidor Meteor se codificó en Node. El manejo de rutas y REST desarrollado en Iron se migró a Express. En cuanto a las colecciones de datos se mantuvieron en Mongo.

Otro desafío que se presentó, fue comunicar el proceso servidor ejecutandose en Raspberry (como un proceso en Raspian) con las placas Arduino Mega y Arduino Nano. Dentro de los paquetes disponibles en NPM, se encontraron dos librerías estables para la comunicación de Node y Arduino. Estas librerías son Cylon[[2]](#footnote-1) y Johnny-five. La librería Cylon utiliza el paradigma de programación declarativo, en cambio, Johnny-five el orientado a *callbacks*. Este último fue el seleccionado por mantener el mismo estilo de codificación que el stack MEAN, compatibilidad con los componentes de Arduino (gracias a estar basado en firmata) y, por sobre todo, poseer una versión estable de serialport compatible con la arquitectura ARM. El protocolo Firmata, base de Johnny-five se analizó en el Capítulo 7 – Librería Johnny-five y el protocolo Firmata).

Finalmente se agregó el código necesario para que las mediciones tomadas por Jhony-Five de los sensores sean almacenados en colecciones de MongoDB. A partir de éstas se generan las estadísticas requeridas por los objetivos de esta tesina.

## **Resumen**

En este capítulo se analizaron las diversas tecnologías tanto de hardware como de software utilizadas en el SAR, justificando la selección de cada una de ellas y los ensayos realizados para concluir en su utilización.

Se explicó el porqué del uso de Arduino y Raspberry Pi como plataformas de base para la manipulación del robot móvil. Las ventajas del uso de la cámara v2 de Raspberry, y las problemáticas que se presentaron al probar la cámara para Arduino OV7670.

Por otro lado, en cuanto al software seleccionado, se detallaron los requerimientos necesarios para el desarrollo del SAR. Dentro de los mismos, se destaca el aprovechamiento de las capacidades brindadas por el Sistema Operativo de Raspberry, seleccionándose Raspbian para tal fin. Finalmente se describen las diversas tecnologías que se probaron a lo largo del desarrollo, resultando MEAN el satck elegido para realizar la aplicación web.

1. <https://www.raspberrypi.org/downloads/raspbian/> [↑](#endnote-ref-1)
2. [↑](#footnote-ref-1)